PRODUCT
DOCUMENTATION

Partial Dependence Plots

BigML Inc.©
2851 NW 9th,
Corvallis, OR 97330, U.S.
https://bigml.com
info@bigml.com

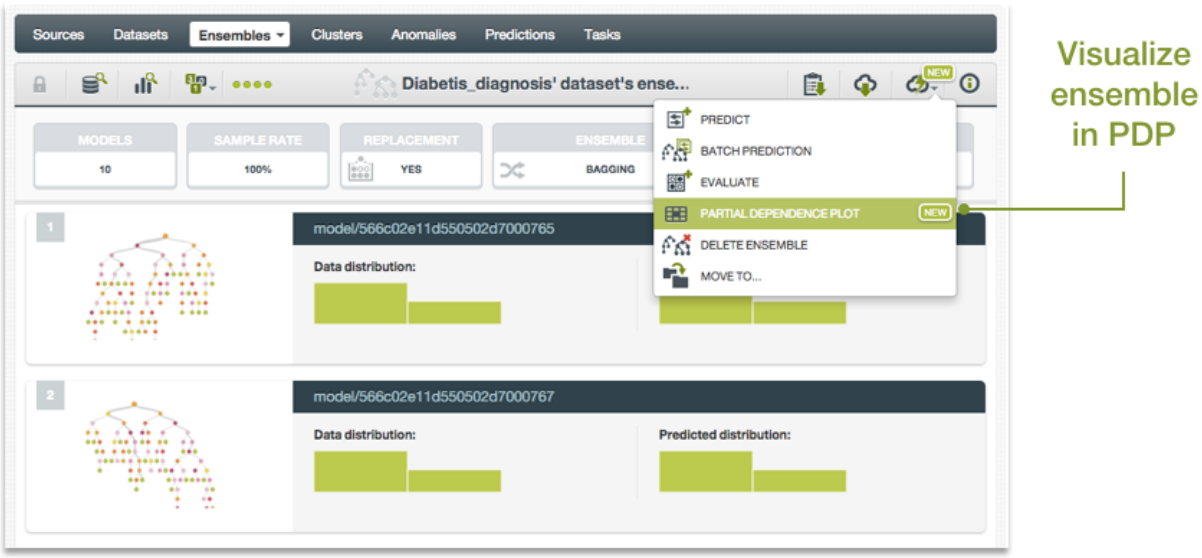
BigML Inc. © 2015

1. PARTIAL DEPENDENCE PLOTS OVERVIEW

Partial Dependence Plot (PDP) is a graphical representation of the marginal effect that a set of variables (predictors) have on the target field (ensemble predictions) ignoring the rest of variables. It is a popular method to interpret the impact of the variables on ensembles predictions and it can be used for classification and regression ensembles. It's very important to notice that PDP is not a representation of the dataset values, it is a representation of the ensemble results.

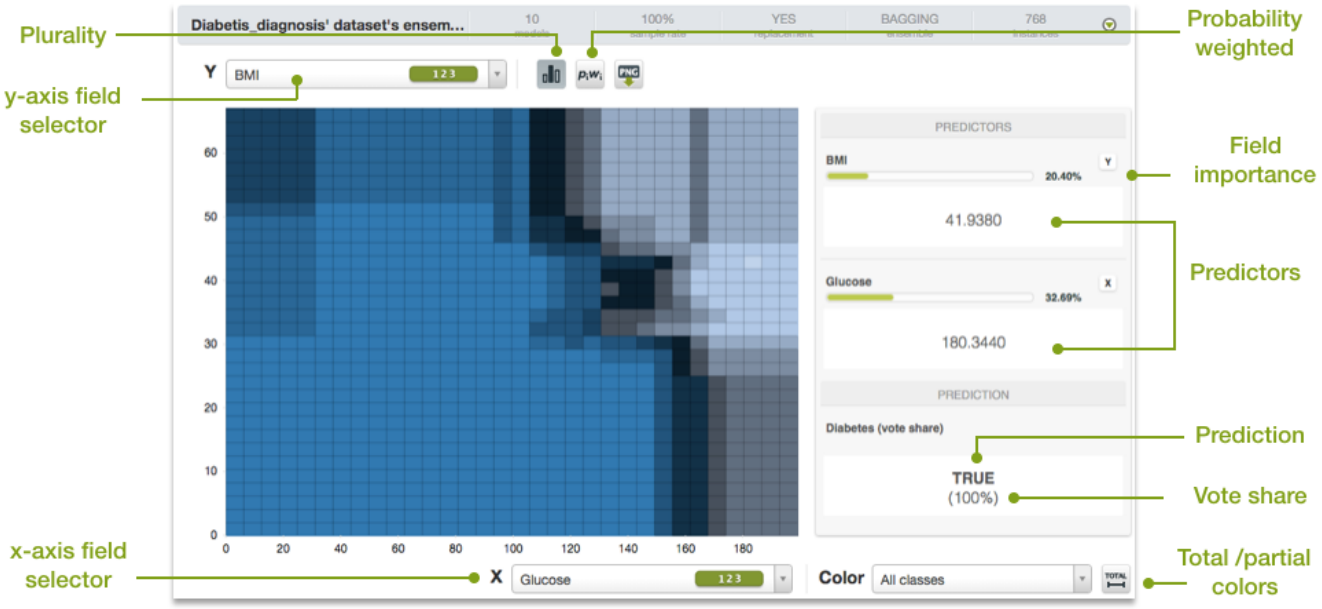
2. ACCESS PARTIAL DEPENDENCE PLOTS

You can access to the PDP from the ensemble results in your Dashboard or from the Labs section by selecting an ensemble.



3. GRAPHICAL REPRESENTATION AND OPTIONS

BigML provides a two-way Partial Dependence Plot where users can select the variables on both axis. Ensemble predictions are represented by a heatmap color chart. Different color tones suggest differences in predictions. Different color shadings for classification ensembles represent different vote shares of classes (see vote share description below).



- **Fields selectors for axis**: you can select the fields for both axis. Fields are ordered in each selector by their importance on predictions (more impactful fields first). You can't select the same field for both axis at the same time. PDP doesn't support text, date-time and items fields (so they can't be selectable in the axis).
- **Color selector for target field**: for classification ensembles you can choose to visualize all classes or just one of the predicted classes.
- **Total / partial colors**: this option allows you to see the color scale and the color shading as dynamic or static.
    - For regression ensembles, it allows you to see the **color scale** for the total range of predictions or relative to the predictions range shown in the chart. For example, consider a regression ensemble for which you have prediction values ranging from 100 to 10,000. Then imagine that for the fields selected in both

BigML Inc. © 2015

axis you just get prediction values ranging from 800 to 850. In this case, if we take into account the whole range of potential prediction values, differences in color won't be perceivable as there are not significant differences in predictions for the fields selected. However if the chart is coloured just taking into account the predictions range shown (800 - 850) we will be able to notice differences in color, although theses differences are not significant for the whole range of potential predictions values (100 - 10,000).

- For classification models, it allows you to see the **color shading** for the total range of potential vote share (from 0% to 100%) or just taking into account the vote share range showed in the chart for the fields selected.

By default, BigML set color scale and shading taking into account the relative range of values shown in chart. If you click the total range icon you will see the chart coloured by the total range of predictions.

- **Plurality**: this voting strategy counts each model prediction as one vote; for classification ensembles the final prediction is the class that has more votes (the mode) and for regression ensembles it is the average of the predicted values.
- **Probability weighted**: this voting strategy uses the average of the probability for each of the classes across trees to select the predicted class. It is only available for classification ensembles due to the fact that it gives the same results than plurality for regression ensembles.
- **Data inspector**: when mousing over the chart surface you will be able to see the **predictors values** along with their **importance** on predictions and the **prediction** in the data inspector to the right.
- **Vote share**: for classification ensembles you will also get the vote share along with the predicted class. The vote share can be understood as a measure of the class strength according to the votes of the individual models in the ensemble. When you select plurality as your voting strategy, vote share is calculated as the percentage of trees that vote for that class. When you set probability weighted as your voting strategy, the vote share is the average of the probabilities of all trees for the predicted class. For example, imagine that you have two trees and two classes, A and B. Given the predictors values, we get the following probabilities for each class:

BigML Inc. © 2015

	Class A	Class B
Decision Tree 1	0.6	0.4
Decision Tree 2	0.8	0.2

If you set plurality as your voting strategy, the class predicted will be A with a vote share of 100% (because all the trees voted for that class). If you select probability weighted as your voting strategy, the predicted class will be A with a vote share of 70% (result of vote share = (0.6+0.8)/2).

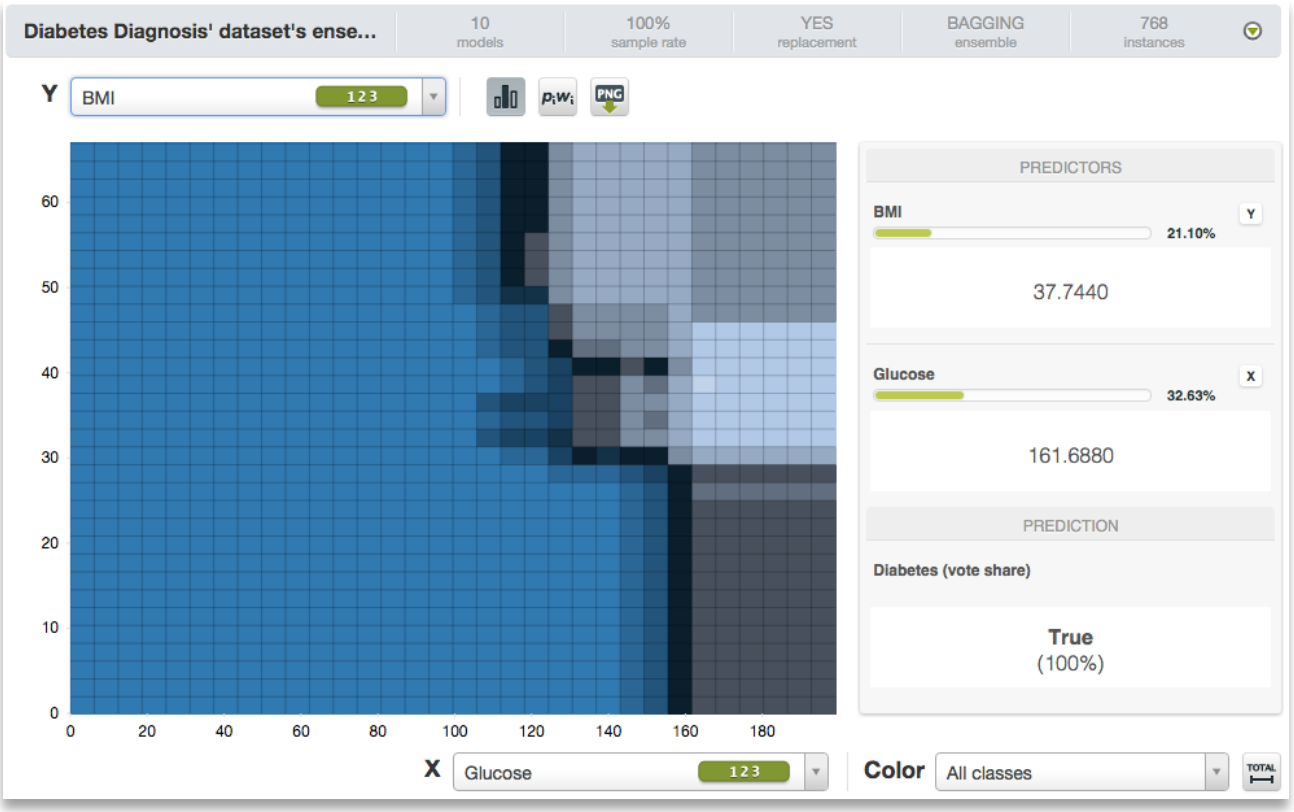
BigML Inc. © 2015

4. INTERPRETING PARTIAL DEPENDENCE PLOTS

You can easily see fields impact on predictions with PDP. Find below three different situations using an ensemble which aims to predict whether a person has or not Diabetes based on several input fields.

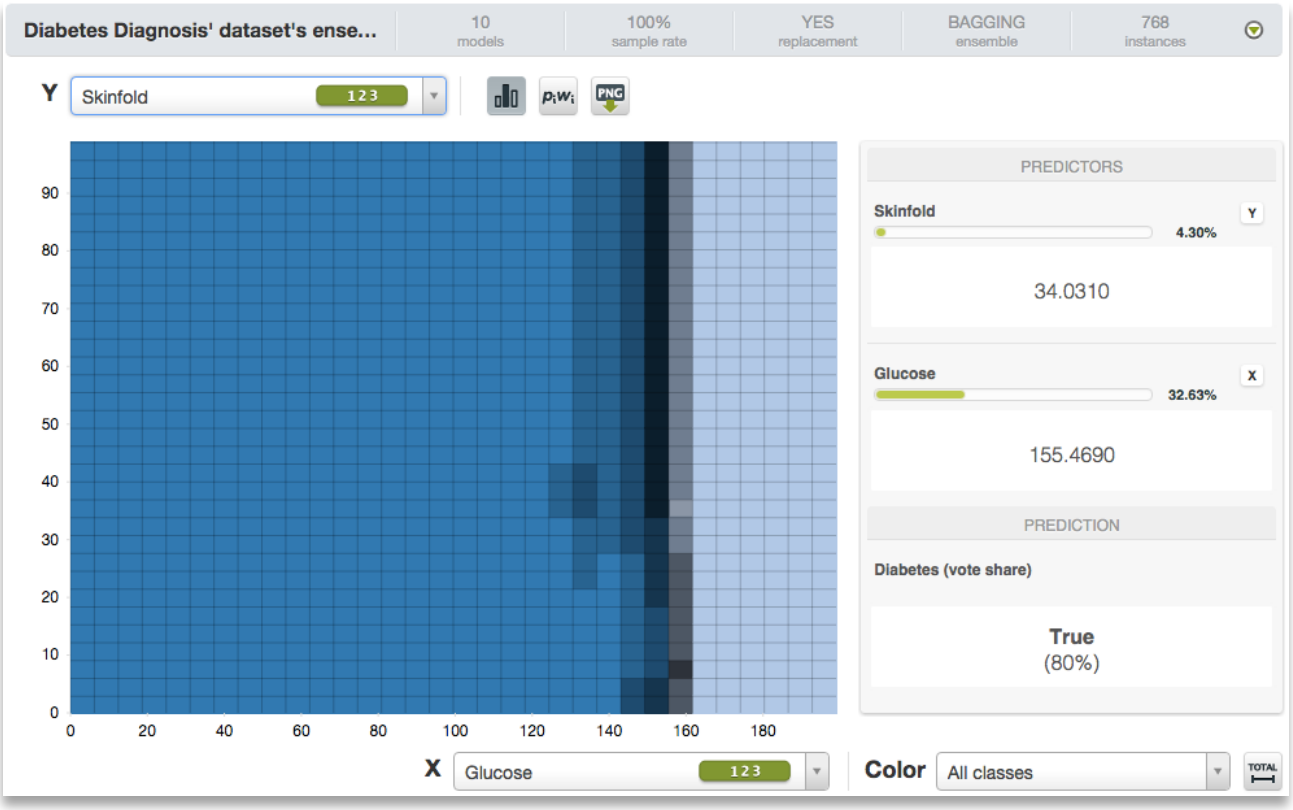
Both fields impact predictions:

In the image below you can see that the combination of the selected fields “BMI” (Body Mass Index) and “Glucose” have a high impact on experiencing Diabetes as variations in both fields cause variations in predictions too.



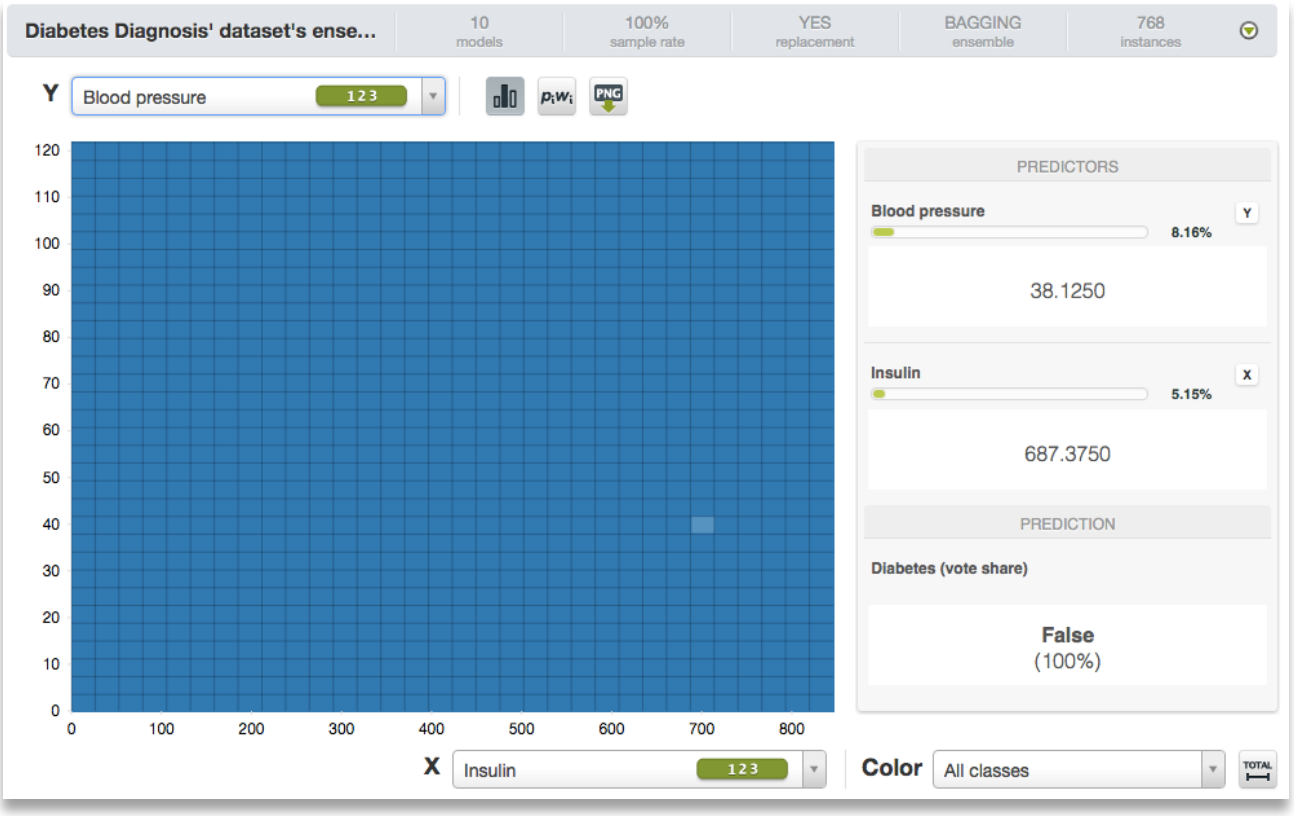
Only one of the fields impact predictions:

Looking at the image below we can conclude that “Skinfold” is not a good predictor for Diabetes as variations in this field don’t affect predictions. On the other hand, we can still see that the level of “Glucose” impacts predictions as we mentioned above.



Both fields have low or not impact on predictions:

If you select variables that have few or no influence on predictions you can see that variations in the selected fields don't lead to differences in predictions. In this case, any combination of "Blood pressure" and "Insulin" always return the same value for Diabetes, "False".




bigml